Scaling Techniques

Why need scaling?

Feature scaling is an important step in preparing data for machine learning, especially for algorithms that are sensitive to the range and distribution of data. It ensures that all features contribute equally to the model, preventing features with larger ranges from dominating the learning process.

Scaling process

In this project, we use Standardization (Z-score normalization) technique to scale the training data and test data. This method scales the data such that each feature has a mean of 0 and a standard deviation of 1.

The reason for choosing the Standardization (Z-score normalization) technique is because most algorithms benefit from this technique:

- Distance-based Algorithms: Algorithms like **k-Nearest Neighbors (k-NN), Support Vector Machines (SVM),** and **k-Means** clustering depend on distances between data points. If the features are normally distributed, Standardization technique is preferred
- Gradient Descent-based Algorithms: Models like **Logistic Regression** and **Neural Networks** benefit from scaled data for faster convergence and performance optimization.

Formula:

$$z = \frac{(x - \mu)}{\sigma}$$

Where:

- x is the original feature value.
- μ is the mean of the feature.
- σ is the standard deviation of the feature.

Code + Output

Scaling features (scaling training data):

We are trying to predict "**Churn_Yes**" column.

We use methods of class **sklearn.preprocessing.StandardScaler()** from Scikit-learn library for standardization.

StandardScaler().fit_transform(X_train) calculates the mean and standard deviation of each feature in the **training data** and then transforms the data by applying the Z-score normalization.

StandardScaler().transform(X_test) transforms the **testing data** by applying the Z-score normalization, based on the previously calculated mean and standard deviation from **fit_transform(X_train)**.

We don't use **fit_transform()** on our testing data because it will indirectly expose our training model to the mean and standard deviation of the testing data, hence, make our training model invalid.

```
from sklearn.preprocessing import StandardScaler


scaler_x = StandardScaler()
scaler_y = StandardScaler()

# Scale the features (X)
X_train_scaled = scaler_x.fit_transform(X_train)
X_test_scaled = scaler_x.transform(X_test)
print("Scaled features result:")
print("--Training set--:")
print(X_train_scaled)
print("--Testing set--:")
print(X_test_scaled)
```

Result:

```
"D:\python\Customer churn rate analysis\.venv\Scripts\python.exe" "D:\python\Customer churn rate analysis\analysis.py"
Scaled features result:
--Training set--:
[[-0.44013836 -0.92130409  0.35446686 ...  1.12563217 -0.51336263
  -0.56431523]
 [-0.44013836  1.56790524  0.40795107 ... -0.88838968 -0.51336263
   1.77205922]
 [-0.44013836 -0.43162357 -1.33362855 ... -0.88838968 -0.51336263
   1.77205922]
 ...
 [-0.44013836 -1.28856449  0.48483463 ...  1.12563217 -0.51336263
  -0.56431523]
 [-0.44013836  0.18047708  0.3327389  ... -0.88838968  1.94794077
  -0.56431523]
 [ 2.2720128  -0.51323699  0.28426884 ...  1.12563217 -0.51336263
  -0.56431523]]
--Testing set--:
[[-0.44013836 -0.43162357  0.4965343  ...  1.12563217 -0.51336263
  -0.56431523]
 [-0.44013836  1.60871195  0.15724383 ... -0.88838968 -0.51336263
   1.77205922]
 [-0.44013836 -0.47243028  0.69877147 ...  1.12563217 -0.51336263
  -0.56431523]
 ...
 [-0.44013836  0.09886366  0.49820568 ...  1.12563217 -0.51336263
  -0.56431523]
 [-0.44013836 -0.79888396  0.3695093  ... -0.88838968 -0.51336263
  -0.56431523]
 [-0.44013836 -1.04372423  0.34109581 ...  1.12563217 -0.51336263
  -0.56431523]]
```

We can transform above NumPy arrays data structures into DataFrame:

```
# Transform scaled features into dataframe for visualization
scaled_X_train_df = pd.DataFrame(X_train_scaled, columns=X.columns)
scaled_X_test_df = pd.DataFrame(X_test_scaled, columns=X.columns)
print("Scaled features DataFrame:")
print("--Training set--:")
print(scaled_X_train_df)
print("--Testing set--")
print(scaled_X_test_df)
```

Result:

```
Scaled features DataFrame:
--Training set--:
      SeniorCitizen    tenure  MonthlyCharges  gender_Male  Dependents_Yes  PhoneService_Yes  MultipleLines_Yes  InternetService_Fiber optic  Contract_One year  Contract_Two year
0         -0.440138 -0.921304        0.354467    -0.994253       -0.658980          0.332933           1.153609                     1.125632          -0.513363          -0.564315
1         -0.440138  1.567905        0.407951    -0.994253       -0.658980          0.332933          -0.866845                    -0.888390          -0.513363           1.772059
2         -0.440138 -0.431624       -1.333629     1.005780       -0.658980          0.332933           1.153609                    -0.888390          -0.513363           1.772059
3         -0.440138 -1.166144        0.947807    -0.994253       -0.658980          0.332933           1.153609                     1.125632          -0.513363          -0.564315
4         -0.440138 -1.002918       -1.504109     1.005780        1.517497          0.332933          -0.866845                    -0.888390          -0.513363           1.772059
...             ...       ...             ...          ...             ...               ...                ...                          ...                ...                ...
5547      -0.440138 -1.288564       -0.688475     1.005780        1.517497          0.332933          -0.866845                    -0.888390          -0.513363          -0.564315
5548       2.272013  1.567905        1.415794     1.005780       -0.658980          0.332933           1.153609                     1.125632          -0.513363          -0.564315
5549      -0.440138 -1.288564        0.484835    -0.994253       -0.658980          0.332933          -0.866845                     1.125632          -0.513363          -0.564315
5550      -0.440138  0.180477        0.332739    -0.994253        1.517497          0.332933          -0.866845                    -0.888390           1.947941          -0.564315
5551       2.272013 -0.513237        0.284269     1.005780       -0.658980          0.332933          -0.866845                     1.125632          -0.513363          -0.564315

[5552 rows x 10 columns]
--Testing set--
      SeniorCitizen    tenure  MonthlyCharges  gender_Male  Dependents_Yes  PhoneService_Yes  MultipleLines_Yes  InternetService_Fiber optic  Contract_One year  Contract_Two year
0         -0.440138 -0.431624        0.496534    -0.994253       -0.658980          0.332933          -0.866845                     1.125632          -0.513363          -0.564315
1         -0.440138  1.608712        0.157244     1.005780        1.517497          0.332933           1.153609                    -0.888390          -0.513363           1.772059
2         -0.440138 -0.472430        0.698771    -0.994253       -0.658980          0.332933          -0.866845                     1.125632          -0.513363          -0.564315
3         -0.440138 -0.554044       -1.480710     1.005780        1.517497          0.332933          -0.866845                    -0.888390          -0.513363           1.772059
4         -0.440138  0.343704        0.539990    -0.994253       -0.658980          0.332933           1.153609                     1.125632          -0.513363          -0.564315
...             ...       ...             ...          ...             ...               ...                ...                          ...                ...                ...
1383      -0.440138 -0.635657       -1.348671    -0.994253       -0.658980          0.332933           1.153609                    -0.888390          -0.513363          -0.564315
1384      -0.440138 -1.288564       -1.502438     1.005780       -0.658980          0.332933          -0.866845                    -0.888390          -0.513363          -0.564315
1385      -0.440138  0.098864        0.498206    -0.994253       -0.658980          0.332933           1.153609                     1.125632          -0.513363          -0.564315
1386      -0.440138 -0.798884        0.369509    -0.994253       -0.658980          0.332933           1.153609                    -0.888390          -0.513363          -0.564315
1387      -0.440138 -1.043724        0.341096    -0.994253       -0.658980          0.332933           1.153609                     1.125632          -0.513363          -0.564315

[1388 rows x 10 columns]
```

Scaling target (scaling testing data):

We can apply the same methods and functions (as done above) to scale our target (testing data).

```python
# Scale the target (y)
y_train_scaled = scaler_y.fit_transform(y_train.values.reshape(-1,1))
y_test_scaled = scaler_y.transform(y_test.values.reshape(-1,1))
print("Scaled target result:")
print("--Training set--:")
print(y_train_scaled)
print("--Testing set--:")
print(y_test_scaled)

# Transform scaled target into dataframe for visualization
scaled_Y_train_df = pd.DataFrame(y_train_scaled, columns=['Scaled_Target'])
scaled_Y_test_df = pd.DataFrame(y_test_scaled, columns=['Scaled_Target'])
print("Scaled target DataFrame:")
print("--Training set--:")
print(scaled_Y_train_df )
print("--Testing set--")
print(scaled_Y_test_df)
```

Note: reshape(-1,1) changes the shape of the array to 2D array. **-1** means “infer the number of rows based on the length of the array” and **1** indicates “one column”. This step is essential because many preprocessing functions such **as fit_transform()** expect the input to be in a 2D array format. Originally, **y_train** (target testing data) is a 1D array, hence, it cannot be input into **fit_transform().**

Result:

```
Scaled target result:
--Training set--:
[[-0.60926412]
 [-0.60926412]
 [-0.60926412]
 ...
 [ 1.6413243 ]
 [-0.60926412]
 [ 1.6413243 ]]
--Testing set--:
[[-0.60926412]
 [-0.60926412]
 [-0.60926412]
 ...
 [-0.60926412]
 [-0.60926412]
 [-0.60926412]]
```

```
Scaled target DataFrame:
--Training set--:
      Scaled_Target
0         -0.609264
1         -0.609264
2         -0.609264
3          1.641324
4         -0.609264
...             ...
5547      -0.609264
5548      -0.609264
5549       1.641324
5550      -0.609264
5551       1.641324

[5552 rows x 1 columns]
--Testing set--
      Scaled_Target
0         -0.609264
1         -0.609264
2         -0.609264
3         -0.609264
4         -0.609264
...             ...
1383      -0.609264
1384      -0.609264
1385      -0.609264
1386      -0.609264
1387      -0.609264

[1388 rows x 1 columns]
```